

Serial No. 08/880,616
Art Unit No. 2151

I. REAL PARTY IN INTEREST

The appeal is made on behalf of Applicants who are real parties in interest with respect to the subject patent application.

II. RELATED APPEALS AND INTERFERENCES

There are no pending related appeals or interferences with respect to the subject patent application.

III. STATUS OF CLAIMS

There are nineteen (19) claims pending in the subject patent application, numbered 1-14 and 16-20. Claim 15 was canceled in the course of prosecution. All of the claims stand rejected.

A complete copy of the claims involved in the appeal is attached hereto.

Serial No. 08/880,616
Art Unit No. 2151

IV. STATUS OF AMENDMENTS

The status of the prosecution of the application is as follows:

December 9, 1998 - Office Action issued rejecting Claims 1-18.

March 9, 1999 - Amendment filed.

June 1, 1999 - Final Office Action issued finally rejecting all claims, 1-18.

September 1, 1999 - CPA with Preliminary Amendment filed.

October 1, 1999 - Office Action rejecting all claims, 1-18.

January 3, 2000 - Amendment filed.

March 15, 2000 - Final Office Action issued finally rejecting all claims, 1-18.

July 17, 2000 - CPA and Preliminary Amendment filed.

October 12, 2000 - Office Action rejecting claims, 1-18.

February 12, 2000 - Amendment filed canceling Claim 15 and adding Claims 19 and 20.

Serial No. 08/880,616
Art Unit No. 2151

May 22, 2001 - Final Office Action rejecting Claims 1-14 and 16-20.

August 22, 2001 - CPA and Preliminary Amendment filed.

October 3, 2001 - Office Action rejecting Claims 1-14 and 16-20.

January 3, 2002 - Amendment filed.

March 28, 2002 - Final Office Action rejecting Claims 1-14 and 16-20.

June 28, 2002 - Notice of Appeal filed.

V. SUMMARY OF INVENTION

The present invention provides a scheduling system and method for a multinode UNIX-based environment. Under the invention, at least one local scheduler prioritizes **processes** in accordance with a global prioritized schedule which is generated at the global scheduler. The local scheduler maintains a local priority list of ready-to-execute tasks correlated with local processes, which list is updated in accordance with the global prioritized schedule provided from the global scheduler. As

Serial No. 08/880,616
Art Unit No. 2151

set forth in the independent claims, the present invention provides a method and system for performing the steps in a UNIX-based environment of scheduling a plurality of tasks of more than one application among processes on at least one computing node, in a system having a global scheduler at at least one computing node having a local scheduler and a plurality of local processes comprising the steps of: dynamically creating a global prioritized schedule of a plurality of tasks, said schedule including tasks of the more than one application; communicating the global prioritized schedule to the computing nodes; determining correspondence between the plurality of tasks and the plurality of local processes; and dynamically prioritizing local processes in a local priority list in accordance with the global prioritized schedule to allow simultaneous execution of tasks from the more than one application. The present approach of global (including inter-node) and local scheduling minimizes unused CPU time when an individual task is temporarily blocked or suspended waiting for I/O.

Serial No. 08/880,616
Art Unit No. 2151

VI. STATEMENT OF ISSUES OF APPEAL

There following issues are on appeal:

(1) whether the teachings of the cited Boland patent (USP 5,872,972) in view of the Applicants' Admitted Prior Art (hereinafter, "AAPA"), obviate the claimed invention as set forth in Claims 1, 11, 12 and 14;

(2) whether the combined teachings of the Boland patent, the AAPA, and the Cameron patent (USP 5,325,526) obviate the invention as set forth in Claims 2, 4-10, 13, 14, 16 and 18-20; and

(3) whether the combined teachings of the Boland patent, the AAPA, the Cameron patent, and the Ripps reference ("The Multitasking Mindset Meets the Operating System") obviate the invention as set forth in Claims 3 and 17.

Serial No. 08/880,616
Art Unit No. 2151

VII. GROUPING OF CLAIMS

The Claims can be considered in one group for purposes of this appeal.

VIII. ARGUMENT

ARGUMENT (1)

The first issue on appeal is (1) whether the teachings of the cited Boland patent (USP 5,872,972) in view of the Applicants' Admitted Prior Art (hereinafter, "AAPA"), obviate the claimed invention as set forth in Claims 1, 11, 12 and 14.

The Boland patent is directed to affinity-based distribution of work in a multiprocessor environment having, in a first embodiment, one node with multiple processors and one scheduler and having, in a second embodiment, multiple nodes and one scheduler. In the first Boland embodiment, with reference to Figs. 2 and 3, the single scheduler 22 queues processes in the order received in a global run queue

Serial No. 08/880,616
Art Unit No. 2151

24. The processes may thereafter be reordered (see: Col. 4, lines 22-24) in the global priority run queue 26 based on process priority. When a processor $P1-Pn$ becomes available to execute a process, the processor examines the queue entries in the global priority run queue 26 to determine if any of the queued processes have affinity with (i.e., were previously run by) that processor. If affinity exists, the processor selects and executes the highest priority affinitized process (steps 36 and 38 of Fig. 3). If no affinity exists, the processor selects and executes the highest priority non-affinitized process (steps 36 and 40).

In the second Boland embodiment, the nodes each have a nodal priority run queue, 71 and 77, for affinitized processes, and there is a global priority run queue, 81, for non-affinitized processes, with scheduler 90 making the affinity determination and directing the processes to the appropriate queue(s). When a processor becomes available, it looks at both queues. How the processor makes a determination as to which process to run is not explicitly taught by Boland. Rather, Boland refers to the teachings of a prior patent application for scheduling in a split transaction bus system. Neither Boland nor the cited prior

Serial No. 08/880,616
Art Unit No. 2151

patent application, however, teach or suggest that the node have a local scheduler which creates a local schedule. Rather, the processor simply executes the next highest priority affinitized process or the next highest priority non-affinitized process when no affinitized processes are waiting, just as in the first embodiment.

Applicants respectfully assert that the Boland patent does not teach or suggest the invention as set forth in the claims. In Boland, there is no local scheduler, no local scheduler prioritized list, no communication of a global prioritized schedule to local nodes, and no updating of a local prioritized list based on the global prioritized schedule. Boland has a FIFO queue at a local node but does not have a local scheduler which maintains a local priority list. Furthermore, Boland does not maintain or update a local priority list, or even update its local queue, based on a global prioritized schedule. No global prioritized schedule is provided to a node under the Boland teachings. Boland simply has a nodal queue which is not altered even if Boland chooses to select a process from the global queue. Under Boland, the available processor choose one of the processes from one of the queues for execution and begins to

Serial No. 08/880,616
Art Unit No. 2151

execute the process. Boland's processor does not and cannot change its queue once a selection is made. Therefore, it cannot be concluded that Boland teaches or suggests the updating of a local priority list in accordance with a global prioritized schedule which is sent to a local scheduler.

The Examiner has cited a phrase from Col. 7, lines 26-28 of the Boland patent wherein it is taught that "these processes [in nodal run queue] may thereafter be reordered based upon process priority within a nodal priority run queues (sic) 71 and 77...". Even if one concludes, based on the foregoing statement, that a local priority sorter exists to sort the FIFO entries from nodal run queue 70 by priority for nodal priority run queue 71, it still cannot be maintained that Boland teaches that a local scheduler is provided for prioritizing in accordance with a global prioritized schedule provided from a global scheduler means. A local sorter would simply sort FIFO entries based on priority, but would not (and under Boland, cannot) update the queue based on a global prioritized schedule (i.e., based on entries from Boland's global priority run queue). Boland does not provide any suggestion of a nodal entity

Serial No. 08/880,616
Art Unit No. 2151

performing those steps. Boland simply discloses that its available processor looks at the two queues, selects a process, and executes the process. Boland does not teach or suggest that a local list or local queue be updated locally based on consultation with the global queue or a global schedule.

Applicants further argue that the cited AAPA does not provide the teachings which are missing from the Boland patent. The AAPA is cited for teaching means for prioritizing processes according to a prioritized schedule. The AAPA teaches assignment of same level priorities to all tasks of an application. Such is not the same or suggestive of prioritizing according to a prioritized schedule. Moreover, even if one were to modify Boland with the AAPA teaching of assigning the same priority level to the tasks of the same application, the combination would not result in the invention as claimed. If same-application tasks are given the same priority level in the Boland queues, that still would not result in a global prioritized schedule being provided to local scheduler for use in updating a local priority list in accordance with the global prioritized schedule.

Serial No. 08/880,616
Art Unit No. 2151

The Examiner next states that "Unix threads/tasks are inherently correlated to the process execution space provided by the scheduling and/or operating system" in rejecting the claim feature of the "means for ascertaining which processes are assigned to tasks" (Claim 1). Applicants first note that the claim language is "means for ascertaining which of said plurality of tasks are assigned tasks, being assigned to each of said plurality of local processes". It appears that the Examiner has equated tasks with local processes (where local processes are defined for the present invention as the local execution components, see: e.g., page 1, line 13 through page 2, line 5 and page 6, line 6 and line 15 with reference to Fig 1.) Applicants have noted this difference in defined terminology to the Examiner throughout prosecution of the present application, yet still find that confusion exists as to the use of the terms. Next, Applicants note that the Examiner's statement itself expressly teaches away from the notion of a global prioritized schedule and the use thereof by a local scheduler. If the tasks are already correlated, then there would be no need for a priority schedule, since it would be predetermined that a local processor would necessarily

Serial No. 08/880,616
Art Unit No. 2151

execute the thread/task regardless of its local schedule and regardless of any global priority. Applicants respectfully assert that the Examiner's statements do not support the use of the AAPA in rejecting the claim language.

The Examiner has also concluded that "it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the local scheduling/correlating means of IBM into the task scheduling system of Cameron, et al...". Applicants respectfully note that the Cameron patent is not part of the rejection of Claims 1, 11, 12, and 14 and therefore have not responded to the Examiner's misstatements. Applicants do, however, address the Cameron patent below with respect to the patentability of Claims 2-10, 13-14 and 16-20.

Finally, in response to the Examiner's statement concerning Claims 12 and 14 that "[t]he operating system would inherently follow any prioritizing scheme employed by the programmer...", Applicants suggest that the Examiner is confusing the two concepts of operating system priorities and scheduler priorities. A scheduler will prioritize based on any number of priority factors such as application requirements, process availability, etc. The scheduler of

Serial No. 08/880,616
Art Unit No. 2151

the present invention also takes into consideration the operating system priorities (e.g., "auto-run virus scan every 48 hours") when implementing its schedule to minimize overrides and interruptions. The Examiner's conclusion that an operating system would inherently follow a programmer's prioritizing scheme would not lead one skilled in the art to modify Boland to invoke operating system priorities when scheduling tasks. If anything, that conclusion seems to teach away from dynamic scheduling in favor of present operating system scheduling. Clearly, such would not lead one to the claimed invention.

For the foregoing reasons, Applicants respectfully conclude that the presently-claimed invention is patentable over a combination of the AAPA and the Boland patent teachings.

ARGUMENTS (2)-(3) will now be addressed.

Applicants will not repeat all of the arguments set forth with regard to the above issues, as all of the claims which are the subject of the second issue on appeal depend directly or indirectly from the independent Claims 1 and 11,

Serial No. 08/880,616
Art Unit No. 2151

but ask that those arguments set forth above be considered in conjunction with the following additional arguments.

ARGUMENT (2)

The second issue on appeal is (2) whether the combined teachings of the Boland patent, the AAPA, and the Cameron patent obviate the invention as set forth in Claims 2, 4-10, 13, 14, 16 and 18-20.

The Examiner has rejected Claims 2, 4-10, 13-16, and 18-20 by adding the Cameron patent teachings to the combination of Boland and the AAPA. Applicants rely on the above-stated arguments that the Boland and AAPA references do not obviate the language of the independent claims 1 and 11, from which Claims 2, 4-10, 13-16 and 18 depend. Moreover, the addition of the Cameron patent does not supply the teachings or suggestions which are missing from the earlier-stated combination.

The Cameron patent is directed to the scheduling of tasks across multiple nodes, wherein a node is defined at Column 2, line 40 of Cameron as a single processor location. Cameron does provide for the execution of parallel tasks of one application on multiple nodes at the same time; however,

Serial No. 08/880,616
Art Unit No. 2151

the present claims call for the scheduling of a plurality of tasks of more than one application, as further detailed below. All of the scheduling and execution actions in Cameron are initiated from the Allocator and Scheduler 710 (see: Figures 6 and 7) which creates partitions, assigns applications to partitions, and creates the schedule for task execution on the partitions using either interactive (i.e., time-slice) scheduling or gang scheduling. The Cameron Allocator and Scheduler creates and directs execution of the global schedule at the local nodes. Applicants point to the teachings found in Column 21, lines 54-67 and from Column 22, line 28 through Col. 23, line 37 of Cameron wherein the Roll Out Partition and Roll In Partition procedures clearly teach that the Allocator and Scheduler controls the execution of the tasks by the partitions. There is nothing in the Cameron patent which teaches or suggests that nodes or partitions have local schedulers which maintain local schedules and which update those local schedules based on the global schedule provided by the Allocator and Scheduler. Rather, Cameron explicitly teaches that a global entity controls the scheduling.

Serial No. 08/880,616
Art Unit No. 2151

Further, the Cameron single level global scheduling includes no means or steps for deciding what process or task should execute when a single process of the currently-scheduled parallel job is suspended or waiting, and therefore, no dynamic creating of a schedule and no dynamically prioritizing of local processes to allow simultaneous execution of tasks from more than one application. Clearly, Cameron does not provide the teachings which are missing from the Boland patent.

With regard to the Examiner's reliance on the fact that Cameron teaches that multiple applications can be assigned to a single processor, Applicants point to the Cameron teachings that only one application can be active and ready to run at a time. (Applicants direct the Board's attention to the statement in Col. 4, line 5 et seq of Cameron that "although more than one application is assigned in a partition, an entire application is scheduled at once across all the nodes on which it is loaded"). Those applications are assigned by the Allocator and Scheduler, which "global scheduler" issues a single directive to execute the scheduled tasks at the nodes of a partition at a particular time. Neither the node nor the partition has a local

Serial No. 08/880,616
Art Unit No. 2151

scheduler and neither has the capability to prioritize the assigned tasks. Moreover, given the explicit Cameron teaching that only one application executes at any given time, the global scheduler of Cameron does not teach or suggest the capability of dynamically assigning tasks of multiple processes in order of importance to utilize idle CPU time. Clearly, therefore, the Cameron patent does not supply the missing teachings to obviate the invention as claimed.

Moreover, Applicants respectfully assert that the Examiner has not logically applied the teachings of the Cameron patent to the Boland and AAPA combination and that any such combination does not render the language of Claims 2, 4-10, 13, 14, 16, 18-20 unpatentable.

With regard to the language of Claim 2, the Examiner has concluded that the Cameron statement "[i]nteractive scheduling using Unix, or other operating systems in a single processor environment, is well known to those of ordinary skill in the art" would serve to obviate the Claim 2 language of "at least one operating system for receiving input from said means for prioritizing and for directing said assigned processes to execute said tasks in accordance

Serial No. 08/880,616
Art Unit No. 2151

with said prioritizing." Applicants respectfully assert that the Cameron patent was expressly teaching away from that prior art approach for a single processor environment, since Cameron seeks to address parallel processing across multiple nodes. Even if one extracted those teachings and applied them to Boland, one would simply have so-called interactive, also known as time-slice, scheduling applied by the Boland schedulers 22 and 90. Clearly such would not obviate the Claim 2 language.

With regard to Claim 4, Applicants respectfully disagree with the Examiner's conclusion that "[s]cheduling information must inherently be obtained by some means in order to produce a prioritized list of tasks." Cameron does not teach or suggest producing a prioritized list of tasks, or communicating a prioritized list of tasks to nodes. Rather, Cameron teaches that its Allocator and Scheduler performs the Schedule Partition procedure detailed in Columns 21-23 and delivers tasks to nodes for execution. Moreover, the Make Partition teachings referred to by the Examiner detail partition mapping and not scheduling per se.

With regard to Claim 5, the Examiner has again concluded that the Cameron mention of multiple applications

Serial No. 08/880,616
Art Unit No. 2151

overrides the explicit Cameron teachings that only one application is scheduled at a time across all nodes on which it is loaded. Applicants respectfully assert that an interpretation which is directly contrary to the express teachings of the Cameron patent simply cannot be imposed on those teachings.

With regard to Claim 6, the Examiner has concluded that the Cameron patent scheduler means dynamically schedules and then communicates the new schedule to a local scheduler. It has been adequately demonstrated that Cameron does not teach a local scheduler. Moreover, while the cited teachings from Col. 9 do discuss dynamically changing priorities at the Allocator and Scheduler, there is nothing in Cameron which teaches or suggests communicating changes to nodes. Prior to actual scheduling (i.e., Roll Out Partition procedure), the partitions and schedules can be changed without any need or capability for communicating those changes to the nodes.

With regard to Claim 7, the cited teachings regarding access to partition data do not teach or suggest a local scheduler or the communication of information to or from that local scheduler. What the cited teachings refer to are access modes for informing the global scheduler (i.e.,

Serial No. 08/880,616
Art Unit No. 2151

Allocator and Scheduler) of application information which may be utilized for making the partitions and the schedule. There is nothing which relates to communicating from the nodes to the Allocator and Scheduler, let alone of communicating from a local scheduler at a node.

With regard to Claim 8, while Cameron may have a timer, it does not have the local scheduler, the means for communicating between a global and a local scheduler, or a timer in connection with those expressly claimed components.

With regard to Claim 9, the cited teachings regarding hash tables for locating partitions do not obviate the claimed invention since Cameron does not have a local scheduler, the means for communicating between a global and a local scheduler, or a table with identities and addresses for local schedulers. A hash table for storing and resolving stored partition information clearly does not obviate the claim language.

With regard to Claim 10, Applicants reiterate that the Cameron patent does not teach a local scheduler or means for communicating a prioritized schedule to a local scheduler.

With regard to Claim 13, Applicants respectfully reiterate the remarks rendered for parallel Claim 6. There

Serial No. 08/880,616
Art Unit No. 2151

is nothing in Cameron which teaches or suggests communicating changes to nodes; and, prior to actual scheduling (i.e., Roll Out Partition procedure), the partitions and schedules can be changed without any means or steps for communicating those changes to the nodes.

With regard to Claim 16, Applicants note that the Cameron recursive scheduling of sub-partitions until all applications have been made active refers to completing cycles of scheduling and executing a single application on all nodes at one time. In contrast, the Claim 16 language includes the language of Claims 11, 12, 14, and 16 thereby encompasses the dynamic creating of a global schedule of multiple applications for more than one process, the communicating of the schedule to a local scheduler, the prioritizing of local processes in accordance with the global schedule, the parallel executing of tasks, the communicating execution information to the global scheduler, and, then, successive scheduling until all tasks have been completed. Clearly Cameron does not teach or suggest all of those claim features.

With regard to Claim 18, Cameron clearly does not obviate the step of the global scheduler dynamically

Serial No. 08/880,616
Art Unit No. 2151

maintaining a list of the nodes in the context of the claimed method set forth in Claims 11 and 13 from which Claim 18 depends. Again, Cameron does not teach the dynamic creating of a global schedule, the communicating of the schedule to a local scheduler, and the prioritizing of local processes in accordance with the global schedule. It cannot then be concluded that Cameron teaches the foregoing in addition to the Claim 18 step of maintaining at least one list.

With regard to Claims 19 and 20, the Examiner states that the Boland patent suggests those steps. Again, Applicants respectfully assert that the combination of Boland, AAPA, and Cameron do not obviate the method steps of the broader claims and clearly could not be said to obviate the claims which depend therefrom and add limitations thereto.

ARGUMENT (3)

The third issue on appeal is whether the combined teachings of the Boland patent, the AAPA, the Cameron patent and the Ripps reference obviate the invention as set forth in Claims 3 and 17. The Examiner has again cited the

Serial No. 08/880,616
Art Unit No. 2151

combined teachings of Boland, AAPA, Cameron, and further cites the Ripps reference.

The Ripps reference simply provides isolated teachings (i.e., not in the context of scheduling of tasks of multiple applications among processes on at least one computing node) regarding an operating system functionality. Clearly, the addition of the Ripps teachings does not provide the instruction or suggestion to arrive at the invention as claimed, since Ripps also does not teach or suggest the use of a global scheduler for creating and communicating a global prioritized schedule to local schedulers at which local prioritized schedules are updated. Applicants again note that the dependent claims contain all of the features of the base claims and add limitations thereto. Since the combined teachings of Boland, AAPA, and Cameron do not obviate the broader claim language, and since Ripps does not provide any of the teachings missing from that combination, it cannot be concluded that the combination with Ripps obviate the language of Claims 3 and 17.

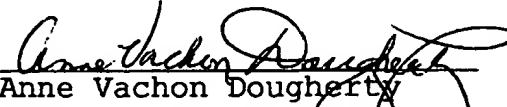
Serial No. 08/880,616
Art Unit No. 2151

CONCLUSION

Applicants respectfully assert that the Examiner has erred in rejecting Claims 1, 11, 12 and 14 as unpatentable over the combined teachings of the Boland patent and the AAPA. Furthermore, Applicants assert that the Examiner has erred in taking features from the Cameron patent and the Ripps reference, and without any suggestion or motivation, inappropriately inserting those teachings into the Boland patent system; and, has erred in concluding that the modification of the Boland system with those features would lead one to arrive at the invention as set forth in Claims 2-10, 13 and 16-20. In light of the foregoing arguments, Applicants request that the decision of the Examiner, rejecting all of the pending claims, be overturned by the Board and that the claims be passed to issuance.

Respectfully submitted,
M.A. COHEN, ET AL

By:


Anne Vachon Dougherty
Registration No. 30,374
Tel. (914) 962-5910

APPENDIX OF CLAIMS

1. Apparatus in a UNIX-based environment for providing scheduling at one time of a plurality of tasks of more than one application among processes in more than one computing node, each node having a plurality of local processes, comprising:

global scheduler means for dynamically creating a global prioritized schedule of said plurality of tasks of said more than one application to allow execution of different tasks of more than one application at the same time at the computing nodes; and

at least one local scheduler associated with each of said more than one computing node comprising means for receiving said global prioritized schedule, means for ascertaining which of said plurality of tasks are assigned tasks, being assigned to each of said plurality of local processes, means for prioritizing said assigned processes, and means to update a local priority list to include said assigned processes in accordance with said global

Serial No. 08/880,616
Art Unit No. 2151

prioritized schedule to allow simultaneous execution of tasks from said more than one application.

2. The apparatus of Claim 1 wherein said at least one computing node additionally comprises at least one operating system for receiving input from said means for prioritizing and for directing said assigned processes to execute said tasks in accordance with said prioritizing.

3. The apparatus of Claim 2 wherein said operating system is further adapted to interleave the execution of local tasks with said tasks.

4. The apparatus of Claim 2 further comprising application coordinator means for communicating information about said plurality of tasks to said scheduler for use in dynamically creating said schedule.

Serial No. 08/880,616
Art Unit No. 2151

5. The apparatus of Claim 2 wherein said local processes are adapted to perform tasks in parallel.

6. The apparatus of Claim 1 wherein said scheduler means comprises global scheduler means comprising means for dynamically scheduling and means for communicating said prioritized schedule to said at least one local scheduler.

7. The apparatus of Claim 6 wherein said local scheduler is adapted to communicate information about said plurality of local processes to said global scheduler.

8. The apparatus of Claim 6 wherein said global scheduler further comprises timer means associated with said communication means to periodically effect communication of said dynamically created prioritized schedule to said local schedulers.

Serial No. 08/880,616
Art Unit No. 2151

9. The apparatus of Claim 6 wherein said global scheduler includes at least one table comprising the identity and address for each of said at least one local scheduler.

10. The apparatus of Claim 2 wherein said scheduler means comprises global scheduler means comprising means for dynamically scheduling and means for communicating said prioritized schedule to said at least one local scheduler.

11. A method in a UNIX-based computing environment for scheduling a plurality of tasks of more than one application among processes on at least one computing node, in a system having a global scheduler means and at least one computing node, each computing node having a local scheduler associated therewith and a plurality of local processes comprising the steps of:

providing application information to said global scheduler means;

Serial No. 08/880,616
Art Unit No. 2151

dynamically creating a global prioritized schedule of said plurality of tasks, said schedule including tasks of said more than one application;

communicating said global prioritized schedule to said more than one computing node;

determining correspondence between said plurality of tasks and said plurality of local processes; and

dynamically prioritizing said local processes in accordance with said global prioritized schedule to allow simultaneous execution of tasks from said more than one application.

12. The method of Claim 11 wherein said dynamically prioritizing comprises invoking operating system priorities to schedule tasks in accordance with said prioritized schedule.

13. The method of Claim 11 wherein said scheduler means is remotely located from said at least one computing node, further comprising the steps of communicating said

Serial No. 08/880,616
Art Unit No. 2151

prioritized schedule of tasks to said at least one computing node.

14. The method of Claim 12 further comprising the step of said local processes executing said tasks in parallel in accordance with said dynamic prioritizing.

16. The method of Claim 14 further comprising the steps of repeating said steps of dynamically creating a prioritized schedule of said plurality of tasks; determining correspondence between said plurality of tasks and said plurality of local processes; and dynamically prioritizing said local processes in accordance with said prioritized schedule; executing; and communicating information about execution until all tasks have been completed.

17. The method of Claim 14 further comprising the step of interleaving execution of local tasks with said executing of said tasks of more than one application.

Serial No. 08/880,616
Art Unit No. 2151

18. The method of Claim 13 further comprising said remotely located scheduler dynamically maintaining at least one list of said at least one computing node.

19. The apparatus of Claim 1 wherein said global scheduler means is adapted to automatically update said local priority list.

20. The method of Claim 11 wherein said dynamically creating a global prioritized schedule of said plurality of tasks comprises the steps of:

receiving task information from at least one of an application coordinator and the more than one computing node;

maintaining an activity scheduler list relating to available processes at said computing nodes and an activity priority list based on said task information.